Problem

- Deriving a variational algorithm generally requires significant model-specific analysis
- These efforts hinder us from quickly developing and exploring a variety of models for a problem at hand.
Goal

Provide a **simple** way to approximate posteriors **quickly** in a **broad** class of latent variable models.

Variational Inference

- A probabilistic model is a joint distribution of hidden variable $z$ and observed variables $x$:
  $$p(z, x)$$

- Inference about the unknowns is through the posterior, the conditional distribution of the hidden variables given the observations:
  $$p(z \mid x) = \frac{p(z, x)}{p(x)}$$

- For most interesting models, the denominator is not tractable. We appeal to approximate the posterior.
Variational Inference

- Variational inference solves posterior computation with optimization
- Define a variational family of distributions over the latent variables $q(z; \nu)$
- Update the parameter to fit the variational distribution to be close (in KL) to the exact posterior

\[ \text{Optimization is over a family } q \]
\[ \text{Find } q \text{ that minimizes } KL(q\|p(z|x)) \]
\[ \text{But KL is intractable; VI optimizes the evidence lower bound (ELBO) instead:} \]
\[ \log p(x) \geq \mathbb{E}_{q_\lambda(z)} [\log p(x, z) - \log q(z)] \]

- **The first term** rewards variational distributions that place high mass on configurations of the latent variable that also explain the observations
- **The second term** encourages variational distribution to be diffuse
- Minimizing the KL is the same as Maximizing the ELBO
Problem

- ELBO:
  \[ \mathcal{L}(\lambda) = \mathbb{E}_{q(z)}[\log p(x, z) - \log q(z)] \]

- Variational inference algorithms are normally derived by computing expectations and gradients
- Expectations often have no closed-form representation
- Computing the required expectations becomes intractable
- Require various expectation for each new model
- How do we get around this?

We want

Black Box Variational Inference
Why do we want Black Box Variational Inference

- Easily use variational inference with any model
- Minimal mathematical work beyond specifying the model $p(z, x)$

General Idea

- Avoid computing model specific expectations and gradients
- Construct noisy gradients by sampling the variational family
Stochastic Optimization

Stochastic optimization maximizes a function using noisy gradients of that function. Formally:

- **Objective:** $f$
- **Parameters:** $x$
- **A noisy gradient** $H; E[H] = \nabla f$
- **Step size:** $\rho_t$
- **Update:** $x_{t+1} \leftarrow x_t + \rho_t H(x_t)$

Converges to a local optimum when:

$$\sum_{t=1}^{\infty} \rho_t = \infty$$

$$\sum_{t=1}^{\infty} \rho_t^2 < \infty$$

A Noisy Gradient of the ELBO

- The ELBO: $\mathcal{L}(\lambda) = \mathbb{E}_{q(\lambda)}[\log p(x, z) - \log q(z)]$

- Gradient of the ELBO:

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \mathbb{E}_q[\nabla_\lambda \log q(z|\lambda)(\log p(x, z) - \log q(z|\lambda))]$$

- How do we construct a noisy gradient of the ELBO?

- Using samples from $q$

- Noisy Gradient:

$$\frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda \log q(z_s|\lambda)(\log p(x, z_s) - \log q(z_s|\lambda)),$$

where $z_s \sim q(z|\lambda)$
A Noisy Gradient of the ELBO

The noisy gradient:

\[
\frac{1}{S} \sum_{s=1}^{S} \nabla_{\lambda} \log q(z_s | \lambda) \left( \log p(x, z_s) - \log q(z_s | \lambda) \right),
\]

where \(z_s \sim q(z | \lambda)\)

To compute the noisy gradient of the ELBO we need

- Sampling from \(q(z)\)
- Evaluating \(\nabla_{\lambda} \log q(z_s | \lambda)\)
- Evaluating \(\log p(x, z)\) and \(\log q(z)\)

A Noisy Gradient of the ELBO

The noisy gradient:

\[
\frac{1}{S} \sum_{s=1}^{S} \nabla_{\lambda} \log q(z_s | \lambda) \left( \log p(x, z_s) - \log q(z_s | \lambda) \right),
\]

where \(z_s \sim q(z | \lambda)\)

Fully black box

- Evaluating \(\log p(x, z)\) is akin to defining the model
- The computations about \(q\) can be shared across models
**Algorithm 1** Black Box Variational Inference

1. **Input:** data $x$, joint distribution $p$, variational family $q$.
2. **Initialize** $\lambda_{1:n}$ randomly, $t = 1$.
3. **repeat**
   4. // Draw $S$ samples from $q$
   5. **for** $s = 1$ to $S$ **do**
   6. $z[s] \sim q$
   7. **end for**
   8. $\rho = t$-th value of a Robbins Monro sequence
   9. $\hat{\lambda} = \frac{1}{S} \sum_{s=1}^{S} \nabla_{\lambda} \log q(z[s]|\lambda)(\log p(x, z[s]) - \log q(z[s]|\lambda))$
10. $\lambda = \lambda + \rho \hat{\lambda}$
11. $t = t + 1$
12. **until** change of held out predictive likelihood is less than $\epsilon$

---

**Variance Control**

- Theoretically this is guaranteed to converge to local optima.
- But unfortunately the variance of the estimator can be very high.
- We have to control the variance of the gradient.
- Two techniques
  - Rao-Blackwellization
  - Control variates
Intuition

- Variance reduction methods work by replacing the function whose expectation is being approximated by Monte Carlo with another function that has the same expectation but smaller variance.
- To estimate $E_q[f]$ via Monte Carlo,
- We compute the empirical average of $\hat{f}$

$$E_q[f] = E_q[\hat{f}]$$
$$\text{Var}_q[f] > \text{Var}_q[\hat{f}]$$

Technique: Rao-Blackwellization

- Replace a random variable with its conditional expectation
- Try to estimate $E[J(X, Y)]$ with Monte Carlo
- Compute $E[J(X, Y)|X]$ then estimate via Monte Carlo
- Proof: Define $\hat{J}(X) = E[J(X, Y)|X]$
- We have $E[\hat{J}(X)] = E[J(X, Y)]$
- $\text{Var}(\hat{J}(X)) = \text{Var}(J(X, Y)) - E[(J(X, Y) - \hat{J}(X))^2]$

Rao-Blackwellize the ith component in a Black box manner

$$\frac{1}{S} \sum_{s=1}^{S} \nabla_{\lambda_i} \log q_i(z_s | \lambda_i) (\log p_i(x, z_s) - \log q_i(z_s | \lambda_i)), $$

where $z_s \sim q(i)(z | \lambda)$. 
Technique: Control Variates

- Replace with $\hat{f}(z) \triangleq f(z) - a(h(z) - E[h(z)])$
- $a$ is a scalar chosen to minimize the variance
- $h$ is a function of our choice
- $E_q[\hat{f}] = E_q[f]$
- $\text{Var}(\hat{f}) = \text{Var}(f) + a^2 \text{Var}(h) - 2a \text{Cov}(f, h)$
- Good $h$ have high correlation with $f$

But in **black box** variational inference

- Set $h$ as $\nabla_{\lambda_i} \log q_i(z_s|\lambda_i)$
- Simply because $E_q[\nabla_{\lambda_i} \log q_i(z_s|\lambda_i)] = 0$ for any $q$

Maintains black box nature of the algorithm

Variance Reduction

![Graph showing variance comparison](image)

**Figure:** Variance comparison between our three estimators. The variance is reduced by several orders of magnitude.
Extension

- Set step size using AdaGrad
- Scalability by subsampling observations
  - Get a monte carlo estimate of $\log p(x, z)$

Experiment

*Figure*: Comparison between Metropolis-Hastings within Gibbs and Black Box Variational Inference on a Gamma-Normal-TS model.
Summary

Black box variational inference only needs

- Log joint of the model $p(z, x)$
- Computations of the variational approximation that can be shared across models